HandNavigator: Hands-on Interaction for Desktop Virtual Reality

Paul Kry, Adeline Pihuit, Adrien Bernhardt, Marie-Paule Cani

HandNavigator: Hands-on Interaction for Desktop Virtual Reality

Paul G. Kry[1]* Adeline Pihuit[2] Adrien Bernhardt[2] Marie-Paule Cani[2]

[1]McGill University, School of Computer Science
[2]Grenoble Universities & CNRS, Jean Kuntzmann Lab, INRIA Grenoble Rhône-Alpes

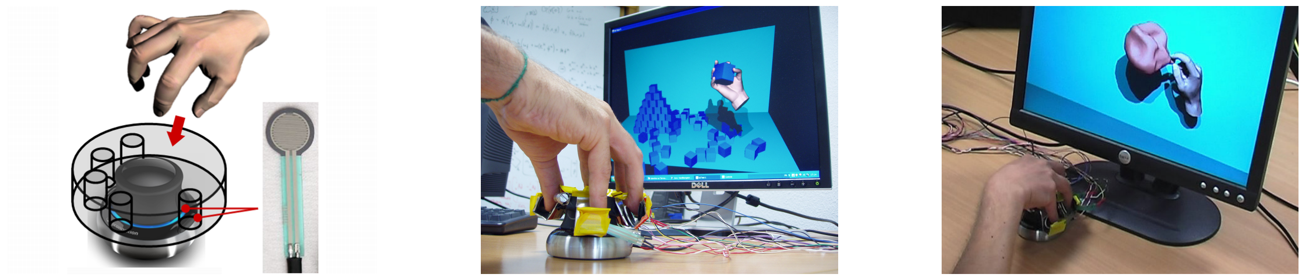

Figure 1: *Left, a conceptual sketch of the HandNavigator device; center and right, examples of hands-on interaction with rigid and deformable virtual environment.*

Abstract

This paper presents a novel interaction system, aimed at hands-on manipulation of digital models through natural hand gestures. Our system is composed of a new physical interaction device coupled with a simulated compliant virtual hand model. The physical interface consists of a SpaceNavigator, augmented with pressure sensors to detect directional forces applied by the user's fingertips. This information controls the position, orientation, and posture of the virtual hand in the same way that the SpaceNavigator uses measured forces to animate a virtual frame. In this manner, user control does not involve fatigue due to reaching gestures or holding a desired hand shape. During contact, the user has a realistic visual feedback in the form of plausible interactions between the virtual hand and its environment. Our device is well suited to any situation where hand gesture, contact, or manipulation tasks need to be performed in virtual. We demonstrate the device in several simple virtual worlds and evaluate it through a series of user studies.

CR Categories: H.5.2 [Information Interfaces]: User Interfaces—Input devices and strategies; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

Keywords: hands, interaction, virtual reality

1 Introduction

In everyday life, our hands are certainly our favorite tool. They perform many different tasks, including gestures, grasping, and other more complex interactions. A task may require specific postures, precise hand positions, and a combination of visual, tactile, and haptic feedback to appreciate the weight, resistance, deformation, and texture of the object or material involved in the interaction.

What about using our own hands for virtual interaction? It has long been a dream to achieve this kind of natural interaction with virtual environments as there are many applications that would benefit, for instance, ergonomics testing for product assembly or disassembly, tele-operation, or risk-free training systems. To a large degree, the dream is already a reality thanks to existing force feedback glove systems. However, these systems can be complex, require careful calibration, and the availability of a large workspace; they are more suited to fully immersive environments than they are to desktop virtual reality environments such as video games, computer aided design, or artistic modeling and sculpting applications.

Bringing convenient hands-on virtual interaction to the desktop is a particularly interesting and challenging problem. This paper presents a novel approach to do this: instead of capturing real hand positions and postures, our new HandNavigator device uses forces exerted by the user's palm and fingers to intuitively control the action of a virtual hand. During interaction, the virtual hand's shape not only depends on the desired gesture, but also on contact with the virtual environment. In particular, it bends naturally thanks to a physical simulation of the contact, providing a realistic visual feedback that is similar to the way a hand would behave in the real world.

1.1 Related work

There exist many solutions for providing hands-on interaction with virtual environments. The most common approach is to control 3D hand models that directly reproduce, in virtual, the position and posture of the user's hands. Different devices can provide this kind of interaction, such as data gloves [Sturman et al. 1989] (see also Immersion's CyberGlove, and Measureand's ShapeHand), motion capture with markers [Pollard and Zordan 2005; Kry and Pai 2006b], or multi-camera stereo with model based tracking [Dewaele et al. 2004]. Other vision based methods, such as [Schlattman and Klein 2007], can improve reliability by restricting the recognition problem to orientation and a small number of gestures. But robust vision based tracking is a very challenging problem, and even

*e-mail: kry@cs.mcgill.ca

with the addition of motion capture markers it is possible to lose track of fingers during grasping due to occlusions. In contrast, active marker tracking is more reliable [Hillebrand et al. 2006], and likewise glove based methods avoid the occlusion problem entirely, though special processing techniques may be necessary to ensure that captured interactions are correct when animated [Boulic et al. 1996].

With approaches based on capturing position, the user makes gestures in the air in the absence of any contact. However, during real interaction, fingers make contact with the environment. This contact can serve as an important support for the hand, and can reduce fatigue associated with holding arm and hand postures. In addition, tactile feedback allows better control of finger movements during an interaction [Johansson 1998], and similarly, it has been shown that performance during precise positioning tasks in virtual environments improves when the user is provided with a device allowing the use of dexterous finger motions with contact [Zhai et al. 1996].

A natural improvement is thus to combine hand motion capture with haptic feedback of virtual interaction forces. Different devices can be used to apply active force feedback on the user's fingers. For example, the SPIDAR [Kim et al. 2000] is an excellent solution for tracking and applying forces on the tip of an individual finger. The Rutgers Hand Master [Bouzit et al. 2002] and the CyberGrasp (Immersion Corporation) provide active feedback to the hand and fingers. While active force feedback has value for many tasks, whole hand force feedback is expensive and complex, and requires the environment-simulation to be capable of producing accurate feedback forces at haptic rates.

An alternative approach is to use a real object as a proxy for an object in the virtual world. In contrast with active force feedback devices, a proxy provides the user with some passive tactile feedback and comes with greatly reduced complexity. As shown by [Insko et al. 2001], passive feedback is a significant improvement over no force feedback at all for immersive virtual environments. Furthermore, proprioceptive senses are easily fooled by visual feedback [Lecuyer et al. 2000]. Previous work on proxies include the Fingerball [Zhai et al. 1996], virtual sculpting with a physical prop [Sheng et al. 2006], and a ball shaped whole-hand interaction device that captures pressures applied by the user's fingers [Pai et al. 2005; Kry and Pai 2006a]. Proxies are useful for simple tasks such as positioning or docking, and additionally (for instance, [Sheng et al. 2006]) as a physical instance of a more complex interface where the deformable prop does not directly represent a virtual object. While a proxy can provide a tangible support for the user's hand and can permit the capture of contact forces, it does not help for position control in the absence of contact.

Input devices such as SpaceBalls and SpaceNavigators have the property of moving very little during use. These devices are known as isometric devices, or as elastic devices in the case where the device's handle has noticeable movement. In contrast, data gloves are known as isotonic devices (they are free-moving and measure position). Elastic and Isometric devices allow control through the measurement of force and torque. The potential for fatigue exists when isometric devices are used for position control [Zhai 1995]; they are more comfortable when used as a rate control, i.e., to control velocity. While position and rate control can be mixed to deal with workspace limits [Dominjon et al. 2006b], our work focuses solely on elastic rate control. An application close to our work is the isometric control of grasping tasks in virtual environments [Kurillo et al. 2007]. However, this system only allows control three fingertips and the position and orientation of the hand and fingers are fixed.

1.2 Contributions

This paper provides a novel alternative to direct position control for hands-on interaction. Instead of capturing the position and posture of the user's hand, the device we introduce captures the multi-directional pressure exerted by the user's palm and fingertips. Pressure measurements are used to control the action of a virtual hand in the same indirect yet intuitive way that a SpaceNavigator controls the position of a virtual frame. We therefore call our new device a *HandNavigator*. Our work brings several important contributions:

- The HandNavigator allows users to control large displacements and arbitrary postures of virtual hands in a desktop setting, using small, natural motions of their own hands.
- The device is easy to use: much like grasping a mouse, there is no setup time for markers, nothing to wear, and passive haptic feedback cues are present.
- It reduces fatigue compared to data gloves: users are not required to hold their arm nor to maintain hand postures in the air; if they do not apply any force, the virtual hand will just maintain its posture. So users can even remove their hand from the device anytime to perform a real-world task and come back to the virtual interaction as they left it.
- Our system can be used to perform a variety of tasks in virtual environments, such as gesturing, touching, and manipulating rigid or deformable bodies. Interaction with virtual objects results in plausible animation of the virtual hand since its configuration not only depends on the desired gesture but also on contact with the environment.
- The HandNavigator is inexpensive to build compared to data gloves and other motion capture systems.

The remainder of this paper first explains the issues and choices made in the design of our new device. Then we present the associated compliant virtual hand model and its control using the HandNavigator for real-time interaction with virtual environments. Finally, we discuss the results of a user study and present directions for future work.

2 Physical interface design

Our focus is the design of an interface allowing people to use their own hand, while applying their actions in a virtual world. We have several important design criteria. First, the physical interface must be easy to use. An important aspect of this is zero setup time (such as when using a mouse): a user should be able to simply place their hand on the device and start working, as opposed to putting on a glove or placing motion capture markers on their fingers. Second, the device must be inexpensive, so as not to restrict its possible general use. Third, it must avoid user fatigue as much as possible, to allow use of the device during long virtual interaction sessions.

Before coming up with our current solution, we created and tested different prototypes. The analysis of their weaknesses motivated the choices that lead us to our final design.

2.1 First prototypes – choices and issues

Our first attempt to control a virtual hand used a Phantom device, a good choice for zero setup time. The position and orientation of the Phantom controlled the local frame attached with the virtual hand, while the buttons available on the stylus controlled preset closing and opening gestures of the hand. This allowed the user to create simple pinching motions (see Figure 2 left). The immediate problem with this set-up was the lack of control of precise hand

Figure 2: *Our first prototype (left) and the second (right), augmenting the Phantom stylus with pressure sensors and a soft ball that serves as a proxy for the virtual objects to be grasped.*

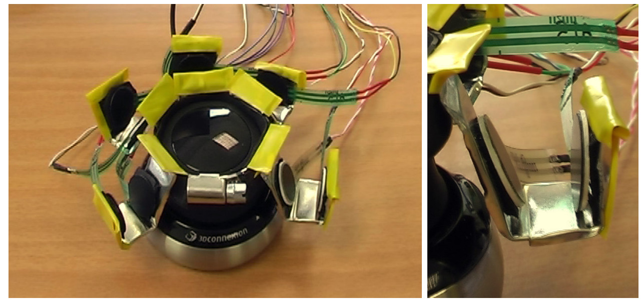

Figure 3: *New prototype left with close-up of one of the FSR sensor petals shown to the right.*

gestures: specifically, the user was not able to control each finger individually, which led to quite restricted types of interaction.

We next improved the device by adding force sensors to capture the pressure exerted by each of the user's fingers, in the manner of the Tango used for interaction capture [Pai et al. 2005]. The force sensors were attached to a soft ball, inspired by the proxy-sponge used in [Sheng et al. 2006]. The soft ball provided the user with some passive haptic and tactile feedback (see Figure 2 right), and measured pressures were used to control the bend of the corresponding virtual finger [Pihuit et al. 2008]. While this second device provided some individual finger control, two main problems were identified:

1. The use of the Phantom to control the position and orientation of the virtual hand resulted in a limited workspace, much reducing the tasks which could be performed in virtual;

2. The pressure sensors on the soft ball provided elastic position control, requiring the user to apply constant finger pressures to hold a constant posture, which was not easy and was tiring.

Despite these problems, this second prototype allowed us to validate a part of our design: forces applied by the user's fingers can successfully activate gestures of the virtual hand. We also observed that holding the real hand in a different posture from the virtual one was not a problem given there was real-time, realistic visual feedback of the interaction.

We thus kept the idea of using finger-pressure sensors for controlling the virtual fingers and designed a dedicated device, which we call the *HandNavigator*, to address the issues above.

2.2 HandNavigator – key ideas

Let us start with the first issue of the limited workspace. Most position input devices do have limited workspaces. For instance, the user cannot go outside the hard limits of the stylus of a Phantom Omni, or outside the magnetic field of a magnetic tracker, outside the range of cameras in optical systems. A common way of working around this problem is to use a clutch, which is analogous to the 2D case where we pick up the mouse and put it down in a new physical location to reach farther positions. Other approaches include scaling the input device workspace to the virtual workspace, or mixing position control with rate control near the boundary of the input device workspace [Dominjon et al. 2006a]. Yet another alternative, however, is to rely entirely on rate control with the advantage being that such devices take up little space and have comparatively less complexity. Consider a SpaceNavigator. Instead of directly controlling the position of a virtual frame, as a mouse would do, the SpaceNavigator remains fixed on the desktop; the user applies directional forces and rotational torques to control the velocity at which the virtual frame translates and rotates in the corresponding direction. The virtual workspace is not limited since the user controls velocity. The handle of the device is not completely immobile, which actually makes it easier to use because small elastic displacements give the user important proprioceptive cues about applied forces and torques. Moreover, when the user stops applying forces (for instance, if the user removes their hand to do something else) then the frame also stops. This can help the user avoid fatigue. Experiments have demonstrated that users can learn relatively quickly this mapping between force and velocity, making a SpaceNavigator a very effective interface for controlling large virtual motions within small desktop environments [Zhai 1995].

Consider now the second issue, namely the control of virtual hand postures. The idea is to control finger postures with forces exerted by the user's fingertips in the same way a SpaceNavigator uses forces and torques. That is, pressure applied by a real finger can control the speed at which the associated virtual finger bends. If the force stops, the virtual finger will just remain in its last position. With such an approach, we also need a mechanism to allow the user to open a finger again, which led us to the idea of providing force sensing 'petals' for fingers. These petals act as holes which receive the user's fingertips, and are each equipped with several force sensors in order to capture pressures applied in different directions. In this way, the user can apply pressure to make a virtual finger bend, but can also apply pressure in the opposite direction to make the virtual finger unfold to its original position.

Naturally, our solution for controlling both the virtual fingers and the position of the virtual hand is to attach these sensor petals to a SpaceNavigator. The petals measure fingertip pressures, while the SpaceNavigator is controlled simultaneously using the entire hand. The combination provides a passive tactile feedback on both the fingers and the hand, and moreover, offers a physical static support for the hand. The device can be oriented on the desktop such that the hand falls comfortably onto the device while the forearm rests on the table. The resulting concept for our HandNavigator is summarized in Figure 1, left.

2.3 Hardware description

Let us now describe the physical prototype we built for our HandNavigator, before detailing in Section 3 how it is coupled with a compliant virtual hand.

Our prototype consists of a SpaceNavigator from 3Dconnexion, and force sensitive resistors (FSRs) to capture fingertip forces. The fingertip pressure sensors are glued to flexible metal elements, which are attached around the sides of the SpaceNavigator's handle much like petals around a flower (see Figure 3 left). The base of each petal has a ridge where the last joint of each finger rests to grasp the

device (see Figure 3 right), while the rest of a petal bends around the fingertip. On the surface of each petal are two sensors for capturing the finger flexion and extension (although we could modify the petals to include lateral pressure sensors to capture additional degrees of freedom). Note that the ridge on each petal allows the user to accurately apply forces on the central SpaceNavigator without applying unwanted pressure on the finger petals sensors. Conversely, thresholds are applied to the SpaceNavigator data readings (more on this below) to allow finger poses to be controlled without creating unwanted hand motion. This way, the user can easily either navigate the hand without moving the fingers, or only move the fingers, or do both at once, as shown in the accompanying video.

The choice of sheet metal for the petals allowed us to bend the petals into different shapes; this was useful in developing comfortable final petal positions. The flexibility of the petals also provides elastic displacement during use, which are a valuable proprioceptive cue when using an isometric rate-control device (similarly, the SpaceNavigator also has a small amount of elasticity in the handle). Note that because of the extra weight of the sensor petals, a recalibration of the SpaceNavigator is necessary to have the device report zero when there is no user action.

Because of the symmetric placement of the petals, the device is easily used with either the left or right hand. However, the software must still know which hand will be used with the device in order to display and control the appropriate virtual hand. Whichever hand, the user places the largest gap between the five petals between the thumb and the little finger, much like a precision grasp on a cylindrical object. Just as different people can grasp the top of a jar with a very similar grip, many different people with hands of different sizes can use our current prototype.

To measure pressure on the FSRs, our prototype uses a data acquisition box from National Instruments. The FSRs are inexpensive and widely available (e.g., from Interlink Electronics). The total cost of building a prototype in this manner is $350, though the device could likely be mass-produced at a fraction of this price (about the same price as a SpaceNavigator).

Both the values measured by the SpaceNavigator and the pressure readings from the FSRs can be used directly as a rate control, i.e., to control the velocity. However, to make the HandNavigator easier to use, we remap the SpaceNavigator signals, specifically, for 2 reasons. The first reason is to create a dead zone around the neutral position to let the user control the fingers or grasp and release the device without causing undesired virtual hand movement. Secondly, we want to give more fine control when the user applies small pressures, and to provide coarse large displacement control when large forces are applied. Our remapping function has 3 easily tunable parameters: the threshold, the one to one mapping position as a percentage of the maximum possible force (minus the threshold), and an exponent. We currently use a threshold equal to one quarter of the maximum possible force, a one to one position at two-thirds, and an exponent of 2 (see Figure 4).

For the FSR petal sensors we do not need such a remapping function since these sensors have a minimum activation pressure (i.e., a built in threshold). Thus, we use direct mapping. Since they are only used intermittently to apply a variable force to move a finger, they do not have problems with hysteresis or drift. That is, any variability in the pressure response is unnoticeable, and the activation threshold is sufficient to prevent finger motion when no forces are applied. Section 3.1 describes in detail how the virtual fingers are controlled via the petal sensors.

Given the parameters described above, and the built in threshold of the petal sensors, the user can control positions and orientations without changing finger positions, and vice versa.

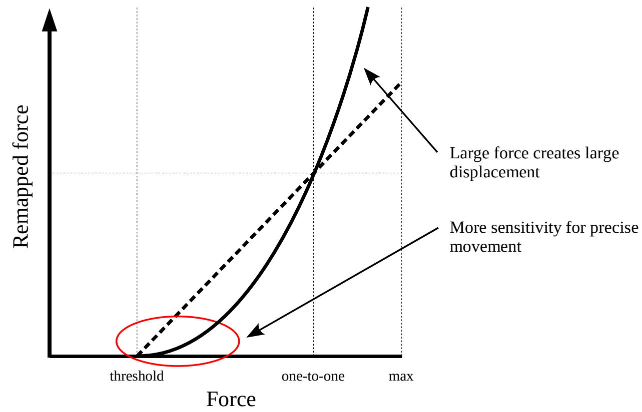

Figure 4: *Remapping function between the force applied on the HandNavigator and the signal used to control the virtual hand.*

3 Virtual interaction

Our physical device represents only half of our new interaction system. To make it practical, we must correctly couple the HandNavigator with a well chosen virtual hand model. The latter should both be controlled by the device and interact in a plausible way with its virtual environment. This involves generating appropriate finger bending responses when virtual contacts occur. Indeed, displaying such plausible interactions in real-time is essential for immersion; realistic visual feedback helps us forget about differences between the actual configuration and forces acting on our own hands and those in the virtual world (note, for example, the pseudo-haptic perception tricks through visual feedback investigated by [Lecuyer et al. 2000]). This allows us to control easily the virtual hand while using a different posture of our own hand, and may even give us the impression of actually touching virtual objects while experiencing a different passive haptic feedback in reality.

3.1 Virtual hand model and finger control

We choose a physically based virtual hand model similar to the one used in [Kry and Pai 2006b]. The hand has 29 degrees of freedom: 6 for position and orientation, 2 in the wrist, 3 in each finger and thumb for flexion and extension, 1 in each finger and thumb for abduction (movement that brings the finger *away from* the medial axis of the hand) and adduction (movement that brings the finger *closer to* the medial axis of the hand), plus 1 more in the thumb for opposition.

To control this hand model, our prototype measures 16 values, 6 of which control position, while the remaining 10 only control flexion and extension. The device does not currently capture the 6 degrees of freedom associated with adduction, abduction, and thumb opposition. Moreover, because we only have one pair of measurement for flexion and extension, we control the 3 joints in each finger in a correlated manner. While this limits the possible hand configurations we can create, the addition of environmental contact forces will allow subtle and complex hand shapes to be produced. Finally, note that we do not try to measure wrist motion, since we associate the rigid motion controlled by the SpaceNavigator with the palm of the hand.

The simulation is quasi-static and there are no masses involved. Instead, only the stiffness of the joints influences how the fingers react with the environment. We use plausible stiffness values based on those estimated in previous work (i.e., [Kry and Pai 2006b]).

To bend a chosen finger, the user presses on the appropriate sensor petal: the inner sensor for flexing the corresponding finger or the outer one for extending it. Nonzero pressures result in proportional joint velocities. To compute the desired finger pose at the next time step, joint velocities are then integrated while taking into account previously selected joint limits, using

$$\theta_{des}^{t+h} = \max(\theta_{min}, \min(\theta_{max}, \theta_{des}^{t} + \dot{\theta}(p)h)) \quad (1)$$

where θ_{des} is the vector of desired joint angles, θ_{min} and θ_{max} give postures of the fully open and fully closed hand, and $\dot{\theta}(p)$ maps FSR pressures p linearly to joint velocities. We use the fully open and closed postures to define these joint velocities. So, sensor values p_{flex} and p_{ext} measuring flexion for a given finger will contribute $k(\theta_{max} - \theta_{min})(p_{flex} - p_{ext})$ to the three joints of that finger. Here k adjusts the overall speed, and can be chosen, for instance, so that the hand goes from fully open to fully closed in a short time (e.g., half a second) at maximum pressure. Note that while it is very difficult to activate opposing sensors at the same time; if it does happen then the joint velocity contributions are simply superimposed.

If there is no contact, the virtual hand will displayed with exactly this integrated pose, while if there is contact, then the desired pose is displaced by contact forces. The quasi-static nature of the model means that there is always equilibrium between the internal torques due to these contact forces and internal torques due to joint displacement. Note that the internal joint torques due to contact forces are easily computed by multiplying the contact force with the transpose of the manipulator Jacobian J for the hand's current position:

$$\theta = \theta_{des} + C\tau \quad (2)$$

with C the compliance (inverse of stiffness), and τ the joint torques defined by

$$\tau = J^T f. \quad (3)$$

We currently do not impose joint limits in the simulation; all the interactions we investigated involved light contacts which do not push the joints beyond their limits. Just the same, minimal and maximal angles (see, [Chalfoun et al. 2003]) could easily be used to constrain the hand to stay within plausible poses during interactions that involve larger contact forces.

Note that a fully dynamic simulation of a virtual hand in the virtual world is also possible in lieu of our quasi-static approach, though a suitable controller would be necessary (e.g., [Pollard and Zordan 2005]).

Finally, we draw a realistic hand in the virtual environment using a standard smooth skinning technique and a model exported from Poser software.

3.2 Controlling virtual hand position and orientation

While mapping the finger petal pressures to virtual finger velocities is straightforward, there are multiple options for using the SpaceNavigator to control the position and orientation of the virtual hand.

The first scenario is that the SpaceNavigator controls the hand position relative to the current view, i.e., pushing the handle towards the left makes the hand move left and pushing to the right makes the hand move right. When controlling a virtual object in 3D, this camera frame mode of interaction can be quite natural as seasoned computer users have deep-rooted intuitions from 2D mouse pointer control.

The second scenario is that the SpaceNavigator controls the hand relative to its own local frame in the virtual environment. That is, pushing down on the SpaceNavigator's handle in the direction of the palm will produce movement of the virtual hand in the direction normal to the palm. If the user sees the back of the virtual hand on screen, then the hand will move farther from the camera; if the palm is visible then the hand will approach the camera. Likewise, bending the hand forward will cause the virtual hand to rotate forward (i.e., around the axis of the thumb). We believe that this is the most logical approach to controlling the virtual hand orientation and position since it acts much like a direct mapping, as if the user's hand was actually located at the current position in the virtual environment. The user feedback we received on this technique is discussed in Section 4.

4 Experimental evaluation

To evaluate the HandNavigator we performed a user study to test the following hypotheses:

- H_1 The user can more quickly prepare to use the HandNavigator (i.e., grasp and release) in comparison to putting on and removing a data glove.
- H_2 Users can control the finger posture of the virtual hand with the HandNavigator with similar ease and speed in comparison to a data glove.
- H_3 Controlling a virtual hand within the hand reference frame is easier than within camera frame.

4.1 Tasks Description

To confirm or refute our hypotheses, we choose a series of tests for each user to perform. These tests can be divided into four categories: moving the virtual hand to different positions and orientations, controlling the virtual fingers, grasping virtual objects, and modeling virtual clay. While first two categories were used for specific criteria of evaluation, the virtual grasping and virtual sculpting tests served as a qualitative evaluation and allowed us to gather spontaneous feedback from users. Each task was designed to help our novice users become more familiar with different capabilities of the HandNavigator. Users were given 5 minutes of training before starting the tasks in each category.

4.1.1 Moving the virtual hand

For each of these tasks in the first category, the user must move the hand to a target position shown with a hand model drawn in a light transparent colour. When the hand is within an error threshold, the user is automatically notified, and shortly thereafter the target is updated to the next position. The first test is a displacement task with fixed orientation of the hand. This allows the user to become accustomed to force based velocity control. The next test is similar to the first, but instead involves only hand rotation. Finally, the user is asked to control both translation and rotation of the virtual hand. All these tests are performed first with a hand-egocentric reference frame, and then with a camera reference frame. Times for completing these tasks were recorded.

4.1.2 Controlling virtual fingers

In the second category of tasks, the user must control the position of the virtual fingers. At first the user must move all the fingers together to open and close the hand, then they are asked to move each finger individually. These tasks are later repeated at the end of the trial using a data glove. This provides a simple comparison of finger control with these two devices. We measured not only the time for individual finger positioning tests, but also the time

necessary to grasp the hand navigator and the time necessary to put on the data glove.

4.1.3 Grasping virtual objects

After a short training step for familiarizing with prehension, the user is asked to grasp some objects of different shapes (cube, bar) in various positions. This requires correctly positioning the virtual hand while flexing virtual fingers. The user is then asked to grasp a cube at the top of a pyramid of cubes, without making the others fall. Finally, the user is asked to push a button with their virtual index finger. These tasks give us an indication of the precision with which a user can control the virtual hand in a dynamic environment.

4.1.4 Modeling virtual clay

In this final step of the evaluation, we use the virtual clay model of [Dewaele and Cani 2004]. The user is free to interact with some virtual clay (i.e., move and deform the clay) and to suggest ideas for improving the device. Modeling clay is a good example of use for the HandNavigator since it requires careful control to edit shapes as desired.

4.2 Experiment setup

The study included 8 individuals (5 males and 3 females) between the ages of 24 and 52, of which 4 had no experience with virtual environments, while the others had familiarly with video games. None of the subjects had previously used a data glove or our device. Among these users, 6 were right-handed and 2 were left-handed.

Tests were designed to give some amount of training to the users and as such were always presented in the same order. During each task we recorded the time to execute this task. We also used a post-trial questionnaire to acquire additional information for a qualitative assessment of criteria such as immersion, fatigue, and precision for the different tasks.

4.3 Experimental results and discussion

From our user study we can first make a general observation regarding the times our subjects needed for achieving the tasks. No difference was noticed according to the gender, handedness, or habits with video games. However, the oldest users were slightly slower.

With respect to hypothesis H_1 our tests showed that putting on the data glove is quite obviously slower than "putting on" the HandNavigator. The average for putting on the data glove and reaching the first hand position was 23 seconds, while the same task was done within 10 seconds with the HandNavigator. Thus for simple tasks, such as adjusting the position of a few fingers, it is actually faster to use the HandNavigator.

Positioning fingers with a data glove is indeed faster than with the HandNavigator, over twice as fast on average for the tasks described in Section 4.1.2 as shown in Figure 5. This suggests that hypothesis H_2 is generally correct, though there are some exceptions. We noticed that subjects using the HandNavigator were slower at opening fingers than for closing them. This may be due to users being unaccustomed to applying pressures with their fingernails, or alternatively that the finger extension sensors could have been better oriented according four our subjects' fingers. Additionally, the outlier in the histogram suggests that the subjects could have benefited from additional training before starting these tasks.

Most of the users were initially disturbed by the hand-egocentric frame. This may be due to the fact that they first considered the HandNavigator as a standard mouse. About ten minutes (according to their own words) were necessary for them to understand the virtual hand behaviour relative to their own hand, after which they were more at ease and performed with times similar to those in the camera frame. Figure 6 shows histograms which combine the times of all users for all positioning tasks of Section 4.1.1. While our hypothesis H_3 may not be correct, users said in informal feedback that they preferred the hand-egocentric frame for precise displacements and rotations (e.g., for grasping).

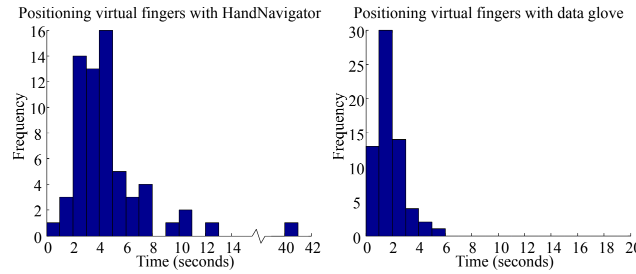

Figure 5: *Histogram of times to perform finger movements with the HandNavigator (left) and with a data glove (right).*

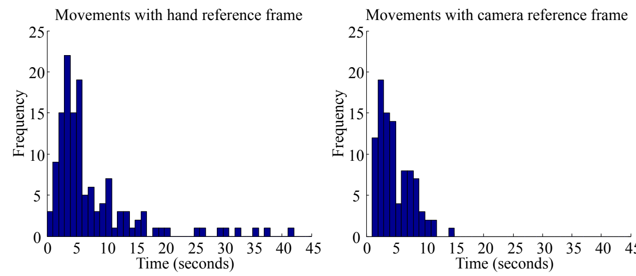

Figure 6: *Histogram of times to perform virtual hand movements using a hand reference frame (left) and a camera frame (right).*

Half of the users did not succeed in the grasping tasks. They explained that it was difficult to evaluate the distance to the object; the farther the virtual hand, the higher the difficulty. This was indeed a very difficult task as we provided neither head motion parallax nor a stereo display. The only depth cues users had were shadows.

Other observations we made during these tests can be summarized as follows. We noted two main advantages: this device is adapted for right-handed as well as left-handed, without distinction; the virtual hand position is kept when the HandNavigator is released (the user can easily take a break and then return to the task later). We also noted a few disadvantages: accurate displacements in the virtual word are difficult when the virtual hand is far from the camera (in part due to the lack of depth cues); coordinating the combined translation of the virtual hand and extension of fingers can be tricky. Finally, it can be difficult to grasp dynamic objects, in part because fine finger control can take slightly longer with rate control.

Finally, let us summarize the comments and observations made by the first users of the HandNavigator. The device is easy to grasp and relatively quick to learn. It can take a user a short while to overcome their initial intuition and realize that they are not holding a mouse (i.e., that pushing left makes the hand move to the hand's left as opposed to left of the screen). Additionally, this mapping of rate control into the local hand coordinates can become difficult to visualize if the virtual hand goes too far from the natural poses we observe of our own hands. For instance, if the right hand is facing

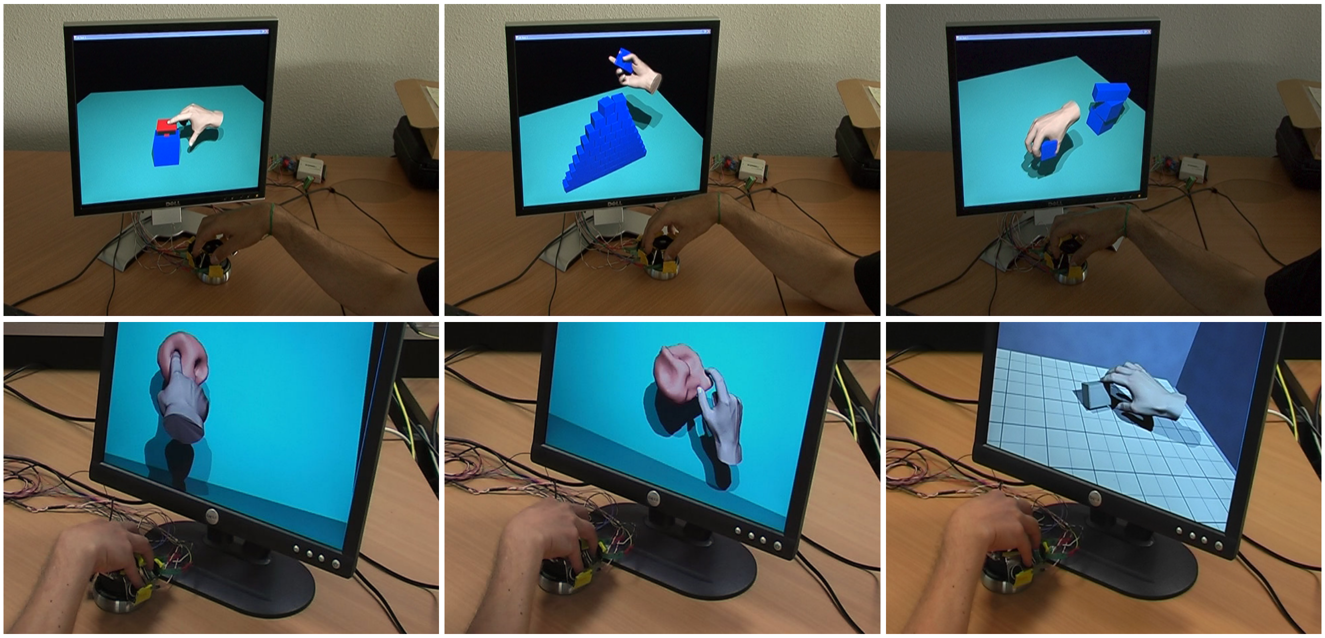

Figure 7: *Example interactions (see video): interacting with rigid objects in a physically based simulation, and interacting with virtual clay, such as making holes and pinching.*

us like the reflection of our left hand then it is possible to lose the intuitive illusion of direct control.

The fact that the HandNavigator only provides control on a restricted family of hand postures was not even noticed by all users. This is possibly because the hand avatar produces complex hand shapes due to contact with the environment, even though our input device only controls flexion and extension of the fingers. This is similar to how we do not have full control over our own hands. Because of how our tendons pull our hands into different configurations, there are many postures that are impossible to obtain without the application of external contact forces. For example, the second and third knuckle in each finger cannot be bent independently, but these joints become independent when we press the fingertip against a solid surface. Furthermore, it has been shown that humans use very few of the degrees of freedom available in their hands when producing imaginary grasps in the air; roughly two degrees of freedom can explain the majority the variation of hand shapes in this case [Santello et al. 1998].

We do not implement pose constraints on the virtual hand. For example, the fourth and fifth fingers are hard to control independently in real life, but could be assigned very different postures using our device; the user could easily apply rate control to different fingers sequentially to produce unnatural hand shapes. Coupling rate control with a biomechanical hand model, such as the one proposed in [Tsang et al. 2005], would be one solution for introducing plausible hand pose constraints.

Figure 7 snapshots of virtual environments used in our experimental evaluation where the HandNavigator is used to interact with rigid and deformable objects (see also the accompanying video).

5 Conclusions

Performing creative tasks in virtual requires intuitive interfaces. One way of making it easier for users to interact is to let them use their own hands within the simulated environment. The challenging case of modeling with virtual clay is just one example of a task that can benefit from such a hands-on interaction. Others include posing virtual hands for keyframe animation, gesturing, pointing, selection through touching, and to some degree, dynamic grasping and manipulation.

Our new device, the HandNavigator, provides a new and inexpensive way for allowing hand-driven glove-free dexterous manipulation in virtual environments. The device is of benefit to anyone looking for a new way to interact with virtual worlds using their own hands in a desktop environment.

The HandNavigator is very inexpensive to build and non-invasive since using it is much like grasping an object. Interaction is based on elastic rate control with a full hand user input device, coupled with a compliant virtual hand simulation. The HandNavigator is a natural extension of a SpaceNavigator; it does not duplicate the physical posture of the user's hand, but offers an intuitive control of a virtual hand while freeing the user from wearing a glove, and the fatigue of holding complex hand postures and repetitive reaching. The physical device supports the user's hand, and the fingertip sensing petals can be comfortably held with a cylindrical precision grasp (similar to a neutral relaxed pose). The device can be used with left or right hands of many different shapes and sizes. Because the user controls a simulated hand avatar that reacts with the surrounding environment, the user is provided visual feedback of plausible posture variations due to complex simulated hand-environment interactions.

5.1 Future work

As mentioned previously, adding extra sensors to the petals of the HandNavigator would allow greater control of the virtual hand. Abduction and adduction are important for creating different kinds of grasps. Additional modifications to capture thumb opposition, however, would be more challenging. We currently provide only visual feedback when there is contact between the virtual hand and the environment. This can be tricky due to occlusion problems. This

could be addressed by altering the response of pressure sensors when contacts occur so that a finger bends less easily, or through the use of vibrations applied at the fingertips to signal the time of contact. These additional cues could facilitate virtual manipulation.

Acknowledgements: This work was supported in part by grants from NSERC and FQRNT. Special thanks to Jérémy Bluteau, Olivier Martin, and Renaud Blanch.

References

BOULIC, R., REZZONICO, S., AND THALMANN, D. 1996. Multi finger manipulation of virtual objects. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, 67–74.

BOUZIT, M., BURDEA, G., POPESCU, G., AND BOIAN, R. 2002. The Rutgers Master II–New design force-feedback glove. *IEEE/ASME Transactions on Mechatronics 7*, 2 (June).

CHALFOUN, J., MASSEMEJEAN, E., MOUHAMED, B., OUESDOU, F. B., AND YOUNÈS, R. 2003. Anatomie de la main et relation tendon couple articulaire. In *Congrès International de Modélisation Numérique Appliquée, CIMNA*.

DEWAELE, G., AND CANI, M.-P. 2004. Virtual clay for direct hand manipulation. In *Eurographics (short papers)*.

DEWAELE, G., DEVERNAY, F., AND HORAUD, R. P. 2004. Hand motion from 3d point trajectories and a smooth surface model. In *Proceedings of the 8th European Conference on Computer Vision*, Springer, T. Pajdla and J. Matas, Eds., vol. I of *LNCS 3021*, 495–507.

DOMINJON, L., LÉCUYER, A., BURKHARDT, J. M., AND RICHIR, S. 2006. A comparison of three techniques to interact in large virtual environments using haptic devices with limited workspace. In *Lecture Notes in Computer Science, Advances in Computer Graphics*, vol. 4035. Springer Verlag, 288–299.

DOMINJON, L., LÉCUYER, A., BURKHARDT, J. M., AND RICHIR, S. 2006. Haptic hybrid rotations: Overcoming hardware rotational limitations of force-feedback devices. In *IEEE International Conference on Virtual Reality (IEEE VR)*.

HILLEBRAND, G., BAUER, M., ACHATZ, K., AND KLINKER, G. 2006. Inverse kinematic infrared optical finger tracking. In *Proceedings of 9th International Conference on Humans and Computers (HC 2006), Aizu, Japan*.

INSKO, B., MEEHAN, M., WHITTON, M., AND JR, F. P. B. 2001. Passive haptics significantly enhances virtual environments. Tech. rep., Computer Science Technical Report 01-010, University of North Carolina, Chapel Hill, NC.

JOHANSSON, R. S. 1998. Sensory input and control of grip. In *Sensory Guidance of Movement. Novartis Foundation Symposium 218*, Wiley & Sons, Chichester, 45–59.

KIM, S., ISHII, M., KOIKE, Y., AND SATO, M. 2000. Development of tension based haptic interface and possibility of its application to virtual reality. In *VRST '00: Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, New York, NY, USA, 199–205.

KRY, P. G., AND PAI, D. K. 2006. Grasp recognition and manipulation with the tango. In *International Symposium on Experimental Robotics, Rio de Janeiro, Brazil, July 6-10*, 9 pages.

KRY, P. G., AND PAI, D. K. 2006. Interaction capture and synthesis. *ACM Transactions on Graphics 25*, 3, 872–880.

KURILLO, G., MIHELJ, M., MUNIH, M., AND BAJD, T. 2007. Multi-fingered grasping and manipulation in virtual environments using an isometric finger device. *Presence: Teleoper. Virtual Environ. 16*, 3, 293–306.

LECUYER, A., COQUILLART, S., KHEDDAR, A., RICHARD, P., AND COIFFET, P. 2000. Pseudo-haptic feedback: Can isometric input devices simulate force feedback? In *VR '00: Proceedings of the IEEE Virtual Reality 2000 Conference*, IEEE Computer Society, Washington, DC, USA, 83.

PAI, D. K., VANDERLOO, E. W., SADHUKHAN, S., AND KRY, P. G. 2005. The tango: A tangible tangoreceptive whole-hand human interface. In *WHC '05: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE Computer Society, Washington, DC, USA, 141–147.

PIHUIT, A., KRY, P., AND CANI, M.-P. 2008. Hands-on virtual clay. In *IEEE International Conference on Shape Modeling and Applications (SMI)*, G. X. Spagnuolo M., Cohen-Or D., Ed., Shape Modeling Interface (SMI), 267–268.

POLLARD, N. S., AND ZORDAN, V. B. 2005. Physically based grasping control from example. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, 311–318.

SANTELLO, M., FLANDERS, M., AND SOECHTING, J. 1998. Postural hand synergies for tool use. In *The Journal of Neuroscience*.

SCHLATTMAN, M., AND KLEIN, R. 2007. Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers. In *VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, ACM, New York, NY, USA, 39–42.

SHENG, J., BALAKRISHNAN, R., AND SINGH, K. 2006. An interface for virtual 3d sculpting via physical proxy. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, ACM Press, New York, NY, USA, 213–220.

STURMAN, D. J., ZELTZER, D., AND PIEPER, S. 1989. Hands-on interaction with virtual environments. In *UIST '89: Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology*, ACM, New York, NY, USA, 19–24.

TSANG, W., SINGH, K., AND FIUME, E. 2005. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, 319–328.

ZHAI, S., MILGRAM, P., AND BUXTON, W. 1996. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of CHI '96*, 308–315.

ZHAI, S. 1995. *Human Performance in Six Degree of Freedom Input Control*. PhD thesis, University of Toronto.